サンプルプログラムについて

１、概要
　このサンプルプログラムはStage3D APIの動作確認と、元になったプログラムの機能拡張を目的に作成された。部分的に元々のプログラムのソースコードが入り交じっているが、Stage3Dの動作を確認するのに丁度良いプログラムです。
　このサンプルプログラムのメインクラスファイルはSceneKitchen.classである。

２、特徴
StringParserクラスを利用することで、テキストファイルからコマンドであるDataPackを作成している。
BoyクラスのActorがActMoveクラスを継承したActWalkクラスのActingを使用することでキャラクターをアニメーションさせている。
Actingのsayを実行したActorにカメラを向けるようにしてある。

３、シナリオファイル
　上に書いたコマンドを生成するテキストファイルのファイル名はscenario.txtである。このファイルはSceneKitchenクラスのinit()メソッド内によって読み込まれ、SceneKitchenクラスのconvertString(String command)メソッドによってコマンドに変換され、実行される。
　それぞれ各一行が一つのActingになるように割り当てられている。テキストは文字"|"によって区切られており、第一句がActingの名前、それ以降がオプションとなるようになっている。それぞれ以下のようになる。

◆ActAddActor

　このActingは新しいActorのインスタンスを作成し、Stageに参加させるものである。

・登録されている名前：add

・構文：

add|$ActorClassName|$ActorName

・引数

$ActorClassName

　登録しようとしているActorのタイプ

$ActorName

　登録するActorの名前

◆ActEnQueue

　このActingは新しいActingのインスタンスを作成し、指定したActorの指定したQueueに追加します。

・登録されている名前：queue

・構文

queue|$Actor|$Num|$Command

・引数

$Actor

　Actingを追加するActorの名前

$Num

　Actingを追加するQueue

$Command

　追加するActingを生成するコマンド。

◆ActInterrupt

　このActingは指定したActorの指定した実行中のActingまたはQueueにたいして割り込みをする。このActingは二つ目の句によって構文が２つに分かれる。構文１はActorの実行中のActingの名前と一致する物にシグナルを投げる。構文２はActorの指定されたQueueの実行中のActingにシグナルを投げる。

・登録されている名前：interrupt

・構文１

interrupt|ACT|$Actor|$Acting|$Signal

・引数

$Actor

　対象となるActorの名前

$Acting

　対象となる実行中のActingの名前

$Signal

　割り込みのシグナル（整数）

・構文２

interrupt|QUE|$Actor|$QueueNum|$Signal

・引数

$Actor

　対象となるActorの名前

$QueueNum

　対象となるQueue

$Signal

　割り込みのシグナル（整数）

◆ActMove

　このActingはActorを移動させるためのものである。これは２つ目の句によって３つの構文に分かれる。構文１は目標となるActorの指定した距離まで近づく。構文２は指定した点まで移動する。構文３は指定した方向に指定した時間の間移動し続ける。

・登録されている名前：move

・構文１

move|Act|$Actor|$Speed|$Distance|$Lock

・引数

$Actor

　移動の目標となるActorの名前

$Speed

　移動の速さ

$Distance

　目標のActorにこの距離より近づいたら移動をやめる。

$Lock

　移動をやめるときActingを終了するかどうか0なら終了する。それ以外なら終了しない。終了しない場合、シグナルを投げないと終了し

ない。

・構文２

move|POS|$X|$Y|$Z|$Speed

・引数

$X　$Y　$Z

移動の目標となる点

$Speed

移動の速さ

・構文３

move|DIR|$X|$Y|$Z|$Speed|$Time

・引数

$X　$Y　$Z

移動する方向

$Speed

移動の速さ

$Time

移動し続ける時間[ms]。0のときは割り込みを受けるまで移動し続ける。

◆ActSay

このActingは発言を行う物である。これは３番目の句によって３つの構文に分かれる。構文１はグループのActorに対して発言する物である（未実装）。構文２は指定したActorに対して発言する物である。構文３は対象なしに発言する物である。

・登録されている名前：say

・構文１

say|$Time|$Volume|GRP|$Group|$Actor|$Speech

・引数

$Time

発言している時間

$Volume

音量

$Group

発言対象のグループ

$Actor

発言対象の（グループの中の）Actor

$Speech

発言内容

・構文２

say|$Time|$Volume|ACT|$Actor|$Speech

・引数

$Time

発言している時間

$Volume
音量
$Actor
発言対象の Actor
$Speech
発言内容

・構文３

say|$Time|$Volume|NOB|$Speech

・引数

$Time
発言している時間
$Volume
音量
$Speech
発言内容

◆ActSetAngle

このActingはActorの方向を設定します。構文１は指定したActorの方向を、構文２は指定した点の位置を、構文３は指定した方向ベクトルの方向を、構文４は方向をlatitudeとlongitudeの二つの角度で指定、構文５はlongitudeのみ設定、構文６はlatitudeのみ設定、である。

・設定されている名前

・構文１

setAngle|ACT|$Actor

・引数

$Actor
目標となるActor

・構文２

setAngle|POS|$X|$Y|$Z

・引数

$X　$Y　$Z
位置ベクトル

・構文３

setAngle|DIR|$X|$Y|$Z

・引数

$X　$Y　$Z
方向ベクトル

・構文４

setAngle|ANG|$Latitude|$Longitude

・引数

$Latitude
Latitudeを指定（360度角）
$Longitude

Longitude を指定（360 度角）

・構文５

setAngle|LON|$Longitude

・引数

$Longitude

Longitude を指定（360 度角）

・構文６

setAngle|LAT|$Latitude

・引数

$Latitude

Latitude を指定（360 度角）

◆ActSetPosition　　setPosition

この Acting は Actor の位置を設定する物である。

・設定されている名前：setAngle

・構文

setAngle|$X|$Y|$Z

・引数

$X　$Y　$Z

設定する位置

◆ActThrowEvent throwEvent

この Acting は受け取った文字列をイベントとして Stage に投げます。そのイベントは最終的に StageCameraMan に渡されます。

・登録されている名前：throwEvent

・構文

throwEvent|$Sentence

・引数

$Sentence

イベントとして Stage に渡される文

◆ActTurn

この Acting は回転の動作を行います。これは２つ目の句により６つの構文に分かれます。構文１は指定した Actor の方向を、構文２は指定した点の位置を、構文３は指定した方向ベクトルの方向を、構文４は方向を latitude と longitude の二つの角度で指定、構文５は longitude のみ設定、構文６は latitude のみ設定、の方向に回転するものである。

・登録されている名前：turn

・構文１

turn|ACT|$Actor|$Speed|$Lock

・引数

$Actor

目標となる Actor

$Speed

回転の速さ（Rotation per second）

$Lock

その方向を向き終わっても終了するかどうか。0のときは終了する。それ以外の時は終了せず割り込みを受けるまで目標となるActorを追従し続ける。

・構文２

turn|POS|$X|$Y|$Z|$Speed|$Lock

・引数

$X　$Y　$Z

位置ベクトル

$Speed

回転の速さ（Rotation per second）

$Lock

その方向を向き終わっても終了するかどうか。0のときは終了する。それ以外の時は終了せず割り込みを受けるまで目標となるActorを追従し続ける。

・構文３

turn|DIR|$X|$Y|$Z|$Speed

・引数

$X　$Y　$Z

方向ベクトル

$Speed

回転の速さ（Rotation per second）

・構文４

turn|ANG|$Latitude|$Longitude|$Speed

・引数

$Latitude
Latitudeを指定（360度角）
$Longitude

Longitudeを指定（360度角）

$Speed

回転の速さ（Rotation per second）

・構文５

turn|LON|$Longitude|$Speed

・引数

$Longitude

Longitudeを指定（360度角）

$Speed

回転の速さ（Rotation per second）

・構文６

turn|LAT|$Latitude|$Speed

・引数

$Latitude
Latitudeを指定（360度角）

$Speed

回転の速さ（Rotation per second）

◆ActWait

このActingはActorを停止させます。これは２つめの句により３つの構文に分かれます。構文１は指定した時間だけ停止、構文２は指定したActorの指定したQueueが空になるまで停止、構文３はの指定したActingが指定したActorのどれかのQueueで実行中である間は停止、である。

・登録されている名前：wait

・構文１

wait|TIM|$Time

・引数

$Time

動作を停止する時間。0の時は割り込みが起こるまで停止する。

・構文２

wait|QUE|$Actor|$QueueNum

・引数

$Actor

待つ相手のActor

$Acting

待つ相手のActorのQueueの番号

・構文３

wait|ACT|$Actor|$Acting

・引数

$Actor

待つ相手のActor

$Acting

待つ相手のActorのActing

◆ActCamSetCamTargetActor

このActingはCamTargetActorのフィールドを設定する。

$Acting = setCamTargetActor

&Command = $Actor

$Actor　：対象となるActorの名前

◆ActCamSetCamTargetPoint

このActingはCamTargetPointのフィールドを設定する。

$Acting = setCamTargetPoint

&Command = $X|$Y|$Z

$X $Y $Z　：ベクトル表す値

◆ActCamSetCamTargetLock

このActingはCamTargetLockのフィールドを設定する。

$Acting = setCamTargetLock

&Command = $Lock

$Lock　：0のときCamTagetLockをfalseに、それ以外のときにtrueを代入。

◆ActCamSetCamViewActor

このActingはCamViewActorのフィールドを設定する。

$Acting = setCamViewActor

&Command = $Actor

$Actor　：対象となるActorの名前

◆ActCamSetCamViewPoint
　　このActingはCamViewPointのフィールドを設定する。]
　　$Acting = setCamViewPoint
　　&Command = $X|$Y|$Z
　　　$X $Y $Z　：ベクトル表す値
◆ActCamSetCamViewLock
　　このActingはCamViewLockのフィールドを設定する。
　　$Acting = setCamViewLock
　　&Command = $Lock
　　　$Lock　：0のときCamTagetLockをfalseに、それ以外のときにtrueを代入。
以上のようになる。

　これらの中でたとえばActAddActorはStageにActorを参加させるActingであるので、これを使って「"Boy”というActorのサブクラスのインスタンスを"boy"という名前でStageに追加するActing」をQueueに追加する場合

　add|Boy|boy

となる。

4:シナリオファイルの書き方

　シナリオファイルはコマンド書きつづった物であり、すべてStageManagerにわたされ解釈されます。たとえば

　　add|Boy|boy

と書いたとき、StageManageのQueueに「BoyというActorのタイプをboyと言う名前でStageに追加する」というActingをQueueに入れます。ただし

　　queue|StageManager|0|add|Boy|boy

とすると、「実行されると「StageManagerにBoyクラスをboyと言う名前で追加する」というActingを追加する」となるので動作順序が変わってくるので注意してください。

　上でActorのboyが追加されたとして、boyに「場所を地点(1, 2, 3)に設定するActing」追加したいとする。このとき

　　setPosition|1|2|3

と書くと、これはStageManagerによって解釈されるため「StageManagerの場所を地点(1, 2, 3)に移動する」という意味になり、これは間違いです。このActingをboyに追加するには次のように書きます。

　　queue|boy|0|setPosition|1|2|3

　次にActorのboyを「場所を地点(1, 2, 3)に設定」したのち「地点(5, 4, 3)に速度１で移動」させたい時を考えます。これは二つのActingを連続して実行になるので次のようになります。

　　queue|boy|0|setPosition|1|2|3
　　queue|boy|0|move|POS|5|4|3|1

これを更に移動した後にベクトル(1,0,0)の方向へ速さ0.5で振り向かせたいなら。

　　queue|boy|0|setPosition|1|2|3
　　queue|boy|0|move|POS|5|4|3|1
　　queue|boy|0|turn|DIR|1|0|0|0.5

とします。